An Artificially Intelligent System for the Automated Issuance of Tornado Warnings in Simulated Convective Storms

DYLAN STEINKRUGER, PAUL MARKOWSKI, AND GEORGE YOUNG

Department of Meteorology and Atmospheric Science, The Pennsylvania State University, University Park, Pennsylvania

(Manuscript received 18 December 2019, in final form 9 July 2020)

ABSTRACT

The utility of employing artificial intelligence (AI) to issue tornado warnings is explored using an ensemble of 128 idealized simulations. Over 700 tornadoes develop within the ensemble of simulations, varying in duration, length, and associated storm mode. Machine-learning models are trained to forecast the temporal and spatial probabilities of tornado formation for a specific lead time. The machine-learning probabilities are used to produce tornado warning decisions for each grid point and lead time. An optimization function is defined, such that warning thresholds are modified to optimize the performance of the AI system on a specified metric (e.g., increased lead time, minimized false alarms, etc.). Using genetic algorithms, multiple AI systems are developed with different optimization functions. The different AI systems yield unique warning output depending on the desired attributes of the optimization function. The effects of the different optimization functions on warning performance are explored. Overall, performance is encouraging and suggests that automated tornado warning guidance is worth exploring with real-time data.

1. Introduction

The concept of a tornado warning has evolved over the past two centuries. As early as the 1880s, an automated tornado warning system was envisioned using telegraph wires to trigger public alerts when winds exceeded 70 miles per hour (Holden 1883; Coleman and Pence 2009). Following the successful tornado forecast by Fawbush and Miller in 1948 (Miller and Crisp 1999), the first tornado forecasts were issued to the public throughout the 1950s (Galway 1989; Doswell et al. 1993). Over the next seven decades, tornado forecasts evolved and transformed into the modern tornado warnings currently issued by the National Weather Service (Coleman et al. 2011).

With increasing observational capabilities and theoretical understanding of tornadoes, warning skill increased over time (Brooks 2004; Brooks and Correia 2018). Part of the increase in skill was due to the

implementation of a national network of Doppler radars throughout the 1990s that provided more information about the current state of potentially tornadic storms. With more data available to forecasters, automated data processing algorithms were developed to rapidly identify convective threats. Products such as the mesocyclone detection algorithm (Stumpf et al. 1998), hail detection algorithm (Witt et al. 1998), and tornado detection algorithm (Mitchell et al. 1998) used human developed rule-based algorithms to inform forecasters of severe weather hazards.

More recently, machine learning (ML) has been introduced as a tool to aid in data processing and assist in the forecasting of severe storms. ML models have been used in the explicit prediction of severe weather hazards including storm longevity (McGovern et al. 2017a, 2019a), severe wind (Marzban and Stumpf 1998; Lagerquist et al. 2017), severe hail (Marzban and Witt 2001; Gagne et al. 2017) and tornadoes (Marzban and Stumpf 1996; Adrianto et al. 2009; McGovern et al. 2019b; Lagerquist et al. 2020; Flora et al. 2020). The advantage of well-trained ML models over human developed algorithms is their ability to self-identify important characteristics in large amounts of data without human interpretation. Because of their ability to interpret data, ML models provide a means for

Supplemental information related to this paper is available at the Journals Online website: <https://doi.org/10.1175/WAF-D-19-0249.s1>.

Corresponding author: Dylan Steinkruger, dus925@psu.edu

identifying data features that are critical to the tornado-genesis process, but hidden in massive amounts of data (McGovern et al. 2014, 2017b). Whereas data mining approaches assist in research, the predictions made by ML models can also be used to aid operational severe storm hazard forecasting. Real-time operational products, such as the Bayesian classifier ProbSevere (Cintineo et al. 2014, 2018), provide probabilistic guidance that a storm will produce severe weather based on observations and model output.

Several questions arise as the role and skill of ML model guidance expands in the warning process. How will human forecasters interpret the output? How will human forecasters handle increased amounts of data from the ML products? The hazardous weather test bed (Kain et al. 2003; Clark et al. 2012) has provided an opportunity to study the “human–machine mix” (Karstens et al. 2018) in the forecasting of severe weather. However, systems such as the German Weather Service’s automated severe convection warning proposal system (James et al. 2018) already exist, and so it seems reasonable to explore the feasibility of automated decision making in the tornado warning decision process.

This paper develops a framework for the automated issuance of tornado warnings in simulated convective storms. The warning system is summarized schematically in Fig. 1. The performance of the warning system suggests warning automation is possible in simulated convective storms and the framework presented may be feasible to explore using real-time, real-life data.

In section 2, idealized simulations are performed and the data are processed. In section 3, the temporal component (i.e., when will a tornado occur) of the warning system is developed (green side of Fig. 1). In section 4, the spatial component (i.e., where will a tornado occur) of the warning system is developed (red side of Fig. 1). In section 5, the temporal and spatial components of the system are combined to issue tornado warnings and the performance of the warning system is presented. Finally, a summary and conclusions are presented in section 6.

2. Data and data processing

As with a human forecaster, the first step in the warning decision process for the artificial intelligence (AI) system is to collect relevant data. In a real-life application of the system, data from multiple sources could include observations (e.g., radar, satellite, surface observations, etc.), model analyses, and model forecasts. Real-life data comes with inherent challenges related to data quality and availability. Therefore, as a first step in examining the feasibility of the proposed system, an ensemble of idealized convective storm simulations was

generated using Cloud Model 1, release 19 (CM1; Bryan and Fritsch 2002). In the context of real-life data, these simulations can be thought of as high-resolution model analyses where diagnostic information about the current state of the storm (e.g., pressure, vorticity, velocity, etc.) and environment (e.g., CAPE, CIN, etc.) are available.

a. Input data

All simulations are performed on a $156 \text{ km} \times 192 \text{ km} \times 18 \text{ km}$ grid. The horizontal grid spacing is 200 m, and the vertical grid is stretched, from a minimum grid spacing of 50 m at the lowest grid level to 250 m above 7 km, resulting in 94 vertical grid levels. The north and south boundaries are periodic; the west and east boundaries are open. The top and bottom boundaries are rigid and free slip. The large and small time steps are adaptive, and each simulation is run for 5 h. Output is saved every 2 min. However, in order to reduce storage needs, output more than 60 min before the first tornado is discarded. In nontornadic simulations or in simulations in which the first tornado does not develop until after 200 min, only the first 140 min are discarded. The domain translates at a constant speed (the speed varies between simulations) to keep storms within the domain. Subgrid-scale turbulence is parameterized using a turbulent kinetic energy (TKE) scheme similar to the one used in Deardorff (1980). A Coriolis acceleration is applied to the wind perturbations from the base-state profile using an f -plane approximation, where $f_0 = 7.5 \times 10^{-5} \text{ s}^{-1}$. Surface fluxes and radiative transfer are not included. The storm initiation methods and microphysics parameterization are explained below, as these vary within the ensemble.

Seven parameters are varied within the ensemble (Table 1). Each parameter uses one of two possible settings; thus, the parameter space contains 128 (2^7) simulations. The choice of parameter settings was guided by two criteria: 1) values needed to be realistic and produce sustained convective storms and 2) the convection needed to be at least marginally tornadic, in at least a part of the parameter space (i.e., a given parameter setting must allow tornadoes to form for at least some combination of parameter settings).

Temperature and relative humidity were initialized using the analytic thermodynamic sounding developed in Weisman and Klemp (1982). The lapse rate and surface temperature of the profile were adjusted to produce a sounding with approximately 2017 J kg^{-1} (Fig. 2a) or 780 J kg^{-1} (Fig. 2b) of surface-based CAPE (SBCAPE). The rate at which the relative humidity declines above 1000 m was also modified to create a moist (Fig. 2a) and dry (Fig. 2b) profile. The moist sounding (Fig. 2a) is similar to the Weisman and Klemp (1982) profile, where

FIG. 1. Schematic summary of the automated tornado warning system.

the relative humidity above the well-mixed layer and below the tropopause is defined as

$$RH(z) = 1 - \frac{1}{2} \left(\frac{z}{z_{tr}} \right)^{1.25}, \quad (1)$$

where, z_{tr} is the height of the tropopause, which is 12 000 m. The dry sounding (Fig. 2b) is modified so that the relative humidity declines at a more rapid rate above 1000 m, such that

$$RH(z) = 1 - \frac{1}{2} \left(\frac{z - 1000 \text{ m}}{z_{tr}} \right)^{0.3}. \quad (2)$$

The kinematic profiles are loosely based on the idealized semicircle hodograph from McCaul and Weisman (2001). Modifications have been made by rotating the hodograph and adding a “tail” to represent deep-layer shear (Fig. 3). The low-level semicircle extends from 0 to 3 km and has a radius of either 8 m s^{-1} (Fig. 3a) or 12 m s^{-1} (Fig. 3b). The upper level shear between 3 and 12 km is approximately 12 m s^{-1} (Fig. 3a) or 22 m s^{-1} (Fig. 3b). Above 12 km, the velocity remains constant. Additionally, the hodograph is rotated to change the rear-to-front flow relative to the north–south lines generated in

the simulations. Low CAPE cases use a hodograph angle of either 20° (Fig. 3a) or 35° . High CAPE cases use a hodograph angle of either 30° or 45° (Fig. 3b).

The storm initiation method and model microphysics were also modified. Each simulation was initiated with either a line thermal or cold block. Random amplitude perturbations were applied within each line thermal or cold block. The line thermal had a maximum potential temperature perturbation of 2.5 K. The cold block was initialized with a minimum temperature perturbation of -3 K . Finally, each simulation used either the Morrison double-moment microphysics scheme with graupel (Morrison et al. 2005, 2009) or the National Severe Storms Laboratory microphysics scheme (Ziegler 1985; Mansell et al. 2010).

TABLE 1. Seven parameters modified for each simulation.

Parameter	Option 1	Option 2
SBCAPE	780 J kg^{-1}	2017 J kg^{-1}
Moisture	Dry	Moist
0–3-km shear	16 m s^{-1}	24 m s^{-1}
3–12-km shear	12 m s^{-1}	22 m s^{-1}
Hodograph orientation relative to north	$20^\circ, 30^\circ$	$35^\circ, 45^\circ$
Initiation method	Line thermal	Cold lock
Microphysics	Morrison	NSSL

FIG. 2. Environmental temperature (red) and dewpoint (green) profiles for (a) high-CAPE, moist simulations and (b) low-CAPE, dry simulations.

A large diversity of convective storm morphologies occurs within the parameter space spanned by the simulations (Figs. 4 and 5). Storm modes vary from linear bowing segments to multicellular storms to semidiscrete supercells.

b. Storm object identification

The warning system narrows the domain it is analyzing by identifying areas of interest where tornadoes are most likely to occur. These areas of interest are

FIG. 3. Environmental hodograph for (a) low-shear, small hodograph angle simulations and (b) high-shear, large hodograph angle simulations. Red dots indicate heights in meters.

FIG. 4. Reflectivity at the lowest model level from a sample of low-CAPE simulations. The black dots represent “tracks” of strong surface vortices. An animation of reflectivity at the lowest model level from a subset of several simulations is available as Fig. S1 in the online supplemental material.

individual storms where characteristics of the storm can be examined to assess the storm’s tornadic potential. This method of storm-based forecasting has shown practicality in forecasting real-life convective hazards (Gagne et al. 2017; Lagerquist et al. 2017). Tornadoes generally form in association with a parent updraft, or region of enhanced vertical motion w and

vertical vorticity ζ . Therefore, updraft helicity (Kain et al. 2008) is chosen to identify storms of interest within the domain.

Updraft helicity (UH) follows the form

$$UH = \int_{0m}^{2000m} w|\zeta| dz, \quad (3)$$

FIG. 5. As in Fig. 4, but for a sample of high-CAPE simulations.

where w is the vertical wind component and ζ is the vertical component of vorticity. The magnitude of ζ is used, such that UH ignores whether an updraft is dominated by cyclonic or anticyclonic rotation, instead it only depends on the collocation of rotation and positive vertical motion. A $2 \text{ km} \times 2 \text{ km}$ maximum filter is applied to the w and ζ fields prior to integration. Additionally, the final UH field is smoothed with a $5 \text{ km} \times 5 \text{ km}$ uniform filter.

Simulations with a linear convective mode produce broad regions of positive w and nonzero ζ . This results in

broad regions where UH is nonzero. To identify localized regions of enhanced UH, multiple rounds of thresholding are performed. As a first step, a low threshold (350 m s^{-2}) is used to identify small, developing storms. Although this low threshold does identify several developing cells, it also commonly results in identifying storms that encompass a large portion of the domain (Fig. 6, panel 1). Therefore, if the storm area exceeds 32 km^2 , the contour threshold is increased by 50 m s^{-2} and the points within the polygon are recontoured. This process is repeated until a polygon's

FIG. 6. Storm object identification. Reflectivity in the background. Black contours with blue dots indicate the storm object. The black contour is the outline of the storm object. The blue dots indicate the center of the storm objects. Numbers inside the arrows indicate the updraft helicity threshold (m s^{-2}) used to contour the image. Note, threshold values are actually increased by 50 m s^{-2} , but for conciseness only every third contour step is shown.

area is reduced below the polygon area threshold or the contour threshold exceeds 1100 m s^{-2} . The end result is localized areas of enhanced UH (Fig. 6, panel 6). Following the definition of previous studies (e.g., Lagerquist et al. 2017), each area of enhanced UH at each 2-min time step is a unique *storm object*.

Storm objects in successive time steps are connected to one another as follows. Storm location at the time being analyzed ($t = 0 \text{ min}$) is predicted using an average persistence of the storm's previous motion at any time steps available between $t = -10 \text{ min}$ and $t = -2 \text{ min}$. The forecasted position of the storm's center is connected to the nearest storm object center identified at $t = 0$ as long as the distance separating the storm object's center and the storm's predicted center is less than 4 km. If multiple storm objects exist within the 4 km radius around the predicted storm location, the storm object with the largest UH value at $t = 0$ is appended to the storm track. Additionally, the storm tracks are extended in order of storm longevity (i.e., the oldest storm at $t = -2 \text{ min}$ searches for a storm object to append first, then the second oldest storm, etc.). If a storm finds no storm object that meets the criteria to append, the storm is classified as “dead” and no future storm objects are appended. Additionally, if a storm object identified at $t = 0 \text{ min}$ is not appended to a storm, it is classified as a new storm and allowed to create its own track.

In general, the methods described above tend to result in the identification of a large number of storm objects. This has the benefit of ensuring most tornado producing storms are identified, but also produces a large number of null (i.e., nontornadic) storm objects that must be filtered (see section 3b). Additionally, the storm tracking algorithm is far from perfect with many discontinuities and “jumps” in the storm tracks with poorer performance generally observed in linear storm modes due to their rapidly varying motion and evolution (Fig. 7). Storm track errors could result in the mislabeling of storms (as tornadic or null), thus improving storm object identification and tracking methods could be sources of future improvement. Nonetheless, the performance of the tracking algorithm is sufficient for our problem. The mean (median) time that a storm is identified before producing its first tornado is 52.8 (48) min.

c. Tornado-like vortex identification

The goal of the AI system is to predict tornadoes; thus, a means to objectively identify tornadic circulations is needed. Recognizing that the intense, small-scale circulations that develop in the simulations are best regarded as “tornado-like vortices” given the 200-m grid spacing, for simplicity they will be referred to as tornadoes hereafter. Grid points at the lowest model (25 m) level having a maximum value of ζ greater than 0.15 s^{-1} and a pressure perturbation p' less than -4 hPa

FIG. 7. Storm tracks for a (a) cellular and (b) linear simulation. Blue lines are the storm tracks. Blue circles are the center of each storm object. Only storms that are identified for longer than 10 min have their storm track drawn. Black triangles identify grid points where a tornado has occurred. Times are for reference. The animations for each set of storm tracks are available as Figs. S2 and S3.

are classified as tornadic. The tornadoes (black dots in Figs. 4 and 5) vary in duration, frequency, and associated storm mode. A *K*-means clustering is performed to further highlight the distribution of tornadoes by storm mode and is presented in appendix A.

To assign tornado events to storm objects, neighboring tornadic grid points at each time step are clustered together. The longest lived storm object within 6 km of each tornado gridpoint cluster is classified as tornadic. If no storm object exists within 6 km, the tornado is not assigned to a storm object. In total, 822 storm objects are assigned a tornadogenesis event (i.e., a tornadic storm

object where the parent storm did not produce a tornado in the previous time step).

d. Data extraction

Once a storm object is identified, information about the storm's characteristics and environment are extracted. Regions in and near the storm that are commonly associated with influencing tornadogenesis are identified and probed for information. Examples include the updraft, cold pool, and near-storm environment. Updraft regions are identified as horizontal areas, at various heights, within a 3-km diameter circle centered on the

maximum value of 0–2-km UH within the storm object. A surface cold pool is defined as the region within the storm object where equivalent potential temperature perturbations θ'_E at the lowest model level are less than -1 K. The environment ahead and behind the storm is also examined through vertical soundings at distances ranging from 2.5 to 50 km away from the center of the storm object, in the storm's current along-track direction. Three-dimensional (3D) objects are also identified to collect information about volumes within the storm. For example, a 3D cold pool is identified by points above an expanded (by 3 or 5 km) 2D projection (black lines in Fig. 6 are the nonexpanded projections) of the storm object with potential temperature perturbations $\theta' < -1$ K. Within each storm subregion, statistical characteristics (e.g., maximum, minimum, 25th and 75th percentiles, mean, etc.) of meteorological variables (e.g., θ , ζ , pressure, etc.) are calculated. Each extracted element (e.g., the 75th percentile of ζ in the updraft at 1000 m above ground level) is classified as a *feature*. Storm objects with missing features are removed from the dataset. Missing data are primarily attributable to small, developing storm objects with weak cold pools that do not meet the θ'_E threshold. Tables 2 and 3 list the features extracted from each simulation.

No information about ζ or p' is explicitly collected at or below 250 m within the storm object. These features are removed because our definition of a tornado includes these variables. However, some collected variables do implicitly include information about ζ below 250 m (e.g., UH). These variables are allowed to remain, as only a small contribution ($<25\%$) of their calculation comes from below 250 m.

3. Temporal warning component

The dataset from the 128 simulations contains over 100 000 storm objects and 700 features. The dataset also identifies the time steps when storm objects produce tornadoes. Therefore, it is possible to identify data patterns that precede the development of a tornado.

a. ML model versus AI system

It is important at this point to define the difference between: “model” and “system.” Model refers to individual ML models trained to answer specific questions, such as “Will this storm produce a tornado in 10–12 minutes?” or “Where will this storm be in 18–20 minutes?” A model produces a single answer to a single question. In contrast, system refers to all of the components developed in this paper. The system, by combining the output from multiple ML models,

TABLE 2. Variables used as input features in the ML models. Statistics calculated for the cold pool variables are the 0th, 50th, 100th percentiles, mean, and standard deviation. LML is lowest model level, RH is relative humidity, SHR is shear, SRH is storm relative helicity, CAPE is convective available potential energy, CIN is convective inhibition, MU is most unstable, and SB is surface based.

Cold pool	Point soundings	Storm
Area	Mean RH for 0–0.25, 0–0.5, 0–1 km	Object area
θ' at LML, 500 m	SHR for 0–0.5, 0–1, 0–3, 0–6 km	Lifetime
θ'_E at LML, 500 m	SRH for 0–1 km MUCAPE MUCIN SBCAPE SBCIN	Motion

produces the final warning products. The output from a ML model is a single component in the AI system. The delineation between these two terms is important in choosing statistics to evaluate their performance. Ultimately, it is the performance of the system that is most important, as the system's output (i.e., the tornado warnings) is provided to the end user.

b. Training ML models

ML models were trained to answer the question: “Will the storm object produce a tornado in _ to _ minutes?” Where the _ are filled with integer values, such as 0–30 or 4–6. Three types of lead-time intervals were tested. The largest interval is a 1-model, 30-min system that uses a single model to decide if a tornado will occur in the next 0–30 min for each storm at each time step. Between 2016 and 2018, the average National Weather Service (NWS) tornado warning length was 34.8 min (<https://verification.nws.noaa.gov/services/public/index.aspx>). Thus, the 30-min system attempts to replicate a “typical” NWS tornado warning. The multimodel systems include a 6-model, 10-min system and a 30-model, 2-min system. The 30-model, 2-min system uses the shortest prediction interval (2 min), in which each model outputs the probability of a tornado at 2-min lead-time intervals out to 1 h.

As a first step, the entire dataset is split into a train and test set. The training set contains 75% (96) of the simulations and is used to train the ML models. The remaining 25% (32) of the simulations test the performance of the models.

Given that only $\sim 1.6\%$ of the storm objects are tornadic, a null event (i.e., one where the storm object will not produce a tornado) is most likely. This imbalance results in ML models learning to always forecast “no tornado” as it results in the greatest number of

TABLE 3. As in Table 2. Updraft variables (except for updraft helicity) are retrieved at the LML and 0.25, 0.50, 0.75, 1, 1.25, 1.50, 2, 2.5, 3.0, 4.0, 5.0, 6.0, and 8.0 km. The statistics calculated for each variable are the 0th, 25th, 50th, 75th, 100th percentiles, mean, and standard deviation. LML is the lowest model level, ζ is vertical vorticity, p is pressure, w is vertical velocity, and MMM indicates variables for which only the minimum (0th percentile), maximum (100th percentile), and mean were extracted.

Updraft	3D variables
Area of overlapping $\zeta > 0.01 \text{ s}^{-1}$ and $w > 10 \text{ m s}^{-1}$	Volume of reflectivity $> 50 \text{ dBZ}$ between LML and 6 km
ζ stretching	Volume of $\zeta > 0.03 \text{ s}^{-1}$ between LML and 6 km
p'	Volume of $\theta' < -1 \text{ K}$ between LML and 3 km
θ'	θ' (MMM) in volume above cold pool between LML and 0.5, 1 km
θ'_E	θ' (MMM) in volume of reflectivity $> 50 \text{ dBZ}$ between LML and 3 km
UH for 0–1, 0–3, 2–5 km	θ' (MMM) in volume of $\theta'_E < -1 \text{ K}$ between LML and 1 km
w	θ'_E (MMM) in volume of $\theta' < -1 \text{ K}$ between LML and 1 km
$w \times \zeta$	
ζ	

correct answers. To rectify this problem, the dataset is balanced by reducing the number of null events. Storms that do not produce tornadoes and do not last more than 10 min are eliminated from the training set because they are easy to identify as nontornadic. The remaining null storm objects are randomly removed until there exists an equal number of null and tornadic storm objects in the training set. Balancing the training set can impact the performance of the ML models, thus other methods (e.g., increasing the weight of the tornadic events, oversampling the tornadic events, etc.) should be investigated in future studies.

After subsampling the nontornadic cases, the number of remaining storm objects is variable depending on the forecast period. In the 2-min system, the number of storm objects available to the ML models for training decreases from ~ 3500 for the 0–2-min model to ~ 2000 for the 58–60-min model. The decrease, in the number of available storm objects, is due to a smaller number of storms being identified at long lead times before a tornado occurs. At the other extreme, the 1-model, 30-min

system that forecasts over a larger 0–30-min interval trains on ~ 15000 storm objects. A clear trade-off exists, such that increasing the temporal resolution of the ML models comes at the cost of reducing the size of the training set and possibly the performance of the ML models.

Three types of ML models were trained: logistic regression (LR), random forests (RF) and gradient-boosted decision trees (GBDT). LR and RF models were developed using Scikit-learn (Pedregosa et al. 2011) and GBDT were developed using XGBoost (Chen and Guestrin 2016). Table 4 lists the hyperparameters tuned in each model and the range of values tested through fivefold cross validation of the training set. The hyperparameters generating the greatest area under the precision-recall curve (AUPRC) were used to train the final model (note, precision-recall plots are synonymous with performance diagrams).

A final model was trained on all the data in the training set and then used to make predictions on the test set. To evaluate model performance on the test set, the maximum value of critical success index (CSI) was

TABLE 4. Hyperparameters and the range of values tested for each model type; N is the number of features.

Model type	Hyperparameter	Min:max range of values
RF	No. of trees	600:850
	Tree depth	3:20
	Max features	$\sqrt{N} - \sqrt{N}/2 : \sqrt{N} + 2\sqrt{N}$
	Split function	Gini or entropy
	Min No. of samples to split a node	2:50
	Min No. of samples to create a leaf	1:20
GBDT	No. of trees	100:1000
	Max depth	1:15
	Learning rate	0.005:0.15
	Percent of instances subsampled for each tree	0.75:1
	Percent of features subsampled for each tree	0.2:0.9
	Gamma (controls node splitting)	0:5
LR	L1 ratio	0:1
LinR	Alpha	0:1
	L1 ratio	0:1

found for each set of predictions. CSI was chosen because it avoids the use of true negatives (i.e., where the model predicts no tornado and there is no tornado), which are the majority class in the testing set (Note: unlike what was done for the training set, the number of tornadoes and nulls is not balanced in the testing set). The training methods described above were repeated eight times with unique cases assigned to the train or test set in each iteration.

Model performance degrades rapidly over the first 10–20 min of forecasted lead time (Fig. 8). Additionally, RF slightly outperforms GBDT and LR in the first 10 min of forecasted lead time. Beyond 10 min, RF and GBDT outperform LR, but there is no appreciable difference in performance between RF and GBDT.

Figure 9 shows the temporal resolution that can be achieved with a 30-model system outputting tornado probabilities at 2-min lead-time intervals. Each line graph indicates the forecasted RF probability of a tornado for a linear (Fig. 9a) and cellular (Fig. 9b) storm. In both events, the probabilities are generally low 20–30 min before tornadogenesis. However, as the tornado event approaches temporally, the models tend to become more confident and an increase in probabilities is observed. In general, the probabilities tend to fluctuate more in nonclassical convective storms. For example, in the linear case (Fig. 9a), the 0–15-min tornado probabilities increase by 30–70 percentage points in only 4 min (between 2248 and 2252 UTC).

c. Making warning decisions

While probabilities provide useful information to forecasters, their interpretation may be unclear to end users who desire a binary decision (e.g., the public, emergency managers, etc.). Thus, it is necessary to convert ML model probabilities into warning decisions.

Figure 10 expands the “blend model” of the schematic in Fig. 1. As a simple example, model probabilities are passed through a decision model that issues a tornado warning if a probability exceeds 50% for a given lead time (Fig. 10a). In the example, model probabilities result in the generation of two warning blocks, one from 0 to 8 min and another from 14 to 18 min. This simple example highlights an ability of the AI system to issue discontinuous warnings. Here, warnings are issued if probabilities exceed 50%; however, the thresholds required to issue a tornado warning can be tuned (Fig. 10b) to a desired performance. To tune performance, we need to identify the statistical measures of performance that are appropriate to our problem.

d. Evaluation statistics

While we have shown the performance of the individual models (Fig. 8), the performance of the warning

FIG. 8. Maximum CSI for each 2-min model as a function of lead time. The solid line represents the mean value and the shading corresponds to the 95% confidence intervals. Confidence intervals are calculated assuming a t distribution and using the mean and standard deviation of the CSI scores calculated for each of the eight test sets. Colors correspond to particular ML model types. Red is random forests, green is gradient boosted decision trees, and blue is logistic regression. The solid black line near 0.02 indicates an approximate score of a model that always predicts a tornado.

system needs to be evaluated by statistics capturing important aspects of the warning performance. The recall of the warning system is calculated using the probability of detection (POD):

$$\text{POD} = \frac{T_{\text{wa}}}{T}, \quad (4)$$

where T_{wa} is the number of tornadoes warned in advance and T is the total number of tornadoes that occur. Note, that POD is based on warnings that are issued prior to tornadogenesis, so warnings that are issued after a storm begins producing a tornado are counted as misses. When all tornadoes have a warning issued prior to tornadogenesis, the numerator equals the denominator and the POD is 1. Additionally, this metric can be made more stringent by requiring a lead-time interval to be met, such as the probability of detection with a lead time greater than 10 min (POD10):

$$\text{POD10} = \frac{T_{\text{wa10}}}{T}, \quad (5)$$

where T_{wa10} is the number of tornadoes warned more than 10 min in advance. The importance of this metric is recognized by considering an example of a warning system that warns all tornadoes in advance, but only issues warnings for the lead-time interval between 0 and 2 min. In this case, the POD of the system is 1; however, the warnings, which provide a maximum lead time of 2 min, have little use as they would be nearly impossible to disseminate to the public.

FIG. 9. Probabilities of a storm producing a tornado at lead times into the future for (a) linear storm system and (b) supercell. Indicated times correspond to the times in Fig. 16. Red bars indicate the future times that the storm will produce a tornado. Animations for each storm are available as Figs. S4 and S5.

The next statistic defines the precision of the warning system. For this, a false alarm time ratio (FATR) is defined as

$$\text{FATR} = \frac{\text{FAT}}{\text{TWT}}, \quad (6)$$

where FAT is the number of minutes with a warning issued and no tornado occurring and TWT is the total number of minutes of all warnings. This term is based on the false alarm ratio; however, because warnings are not issued in distinct polygons, the ratio is modified to only consider the temporal effects of warnings issued by the

FIG. 10. Expansion of the “blend model” in Fig. 1 where warnings are issued (a) whenever the ML output probabilities exceed 50% and (b) whenever the ML output probabilities exceed the optimized warning thresholds.

system. This statistic punishes situations where a significant amount of time passes with a warning issued and no tornado occurring. Thus, a perfect scenario is one where a tornado is always occurring at the time predicted by the warning, such that the numerator is 0. Thus, a perfect FATR score is 0. Ultimately, adding a time component to the false alarm statistic forces the system to trim the temporal size of its warnings.

Finally, a term is defined to quantify the amount of time a warning is issued in advance of a tornado. This statistic is not commonly used to evaluate binary problems, but it is important in assessing the skill of the warning system. Here we use a definition of lead time from [Brooks and Correia \(2018\)](#) where lead time in advance of tornadogenesis (LTA) is defined as

$$LTA = \frac{\sum_{i=1}^{T_{wa}} LT_i}{T_{wa}}, \quad (7)$$

where T_{wa} is the number of tornadoes warned in advance and LT_i is the lead time of the i th tornado warned in advance. As with POD, LTA only evaluates warnings that are issued prior to tornadogenesis.

e. System optimization

With the statistics described above, we are able to evaluate and optimize the performance of the warnings issued by the system. To optimize system performance, genetic algorithm (GA) optimization ([Holland 1975](#); [Goldberg 1989](#)) is used to adjust the warning thresholds.

To optimize the system performance, GA requires a system optimization function (SOF). The SOF is a critical component in achieving desired system performance as GA will maximize the performance of the statistics in the function. To begin, the base form of the CSI expressed in terms of POD and false alarm rate (FAR) is introduced ([Schaefer 1990](#)):

$$CSI = [(POD)^{-1} + (1 - FAR)^{-1} - 1]^{-1}. \quad (8)$$

The FAR term in (8) defines the precision, so it is replaced with FATR. Additionally, we note that not all correct tornado warnings are the same, even when issued prior to tornadogenesis. For example, a tornado warned with 30 min of lead time is better than a tornado warned with 2 min of lead time. So, to account for POD and lead time we adjust the POD term, such that

$$POD_{LT} = \frac{\sum_{i=1}^{T_{wa}} \left(\frac{LT_i}{30} \right)}{T}, \quad (9)$$

where T_{wa} is the number of tornadoes warned in advance, LT_i is the lead time of the i th tornado warned in advance, and T is the total number of tornadoes. Additionally, the summed value in the numerator ($LT_i/30$) is capped so that its value cannot exceed 1. Now, the performance of a correct warning increases with increasing lead time. A tornado warning with a lead time of 15 min has a summed value of 15/30 or 0.5. If all tornadoes are warned with a lead time greater than 30 min, the value of POD_{LT} is 1.

Given the above changes a modified critical success index (MCSI) is defined as

$$MCSI = [(POD_{LT})^{-1} + (1 - FATR)^{-1} - 1]^{-1}. \quad (10)$$

Finally, a minimum performance metric factor (MPMF) is defined to further control the system's performance. Ultimately, any metric can be chosen, and we will show the effects of changing this term on system performance in [section 5](#). For now, we define the MPMF as

$$MPMF = \frac{POD_{LT10} - 0.70}{0.10}, \quad (11)$$

where POD_{LT10} is

$$POD_{LT10} = \frac{\sum_{i=1}^{T_{wa}} \left(\frac{LT_i}{10} \right)}{T}. \quad (12)$$

As in (9), the summed value in the numerator is capped so that its value cannot exceed 1. Additionally, negative values of MPMF are set to 0 and values of MPMF greater than 1 are set to 1. Here, the MPMF decreases rapidly as an increasing number of warnings are issued with less than 10 min of lead time. This additional factor thus encourages the thresholds be set in such a way that warnings are issued well in advance of tornadogenesis. The MPMF is multiplied by MCSI to create the system optimization function (SOF):

$$SOF = MCSI \times MPMF. \quad (13)$$

Thus, as GA optimizes the SOF, it is attempting to minimize the FATR, maximize the POD_{LT} , and keep the MPMF term from having a significant drop in performance.

With a SOF in hand, GA optimization ([Fig. 11](#)) is implemented using the DEAP module in Python ([Fortin et al. 2012](#)). Further details on the application of GA can be found in [appendix B](#). GA is applied to the 30-model, 2-min and 6-model, 10-min systems. Optimization of the 1-model, 30-min system does not use GA, rather the single warning threshold value is raised from 0.4 to 1.0

by 0.01, and the warning threshold is set at the maximum SOF value.

Threshold optimization uses 50% of the test data (12.5% of the full dataset). The remaining 50% of the test data (which was neither used to train ML models nor optimize the warning thresholds) evaluates the system's performance. As before, this process was repeated for each of the eight folds.

4. Spatial warning component

So far, this paper has focused on developing aspects of the AI system related to answering “will/when will a tornado occur” (the green side of Fig. 1). Additionally, the performance metrics related to this question have assumed a perfect prediction of the storm track. However, prediction of the storm track is unlikely to be perfect, particularly as lead time increases.

To forecast storm tracks, an elastic net (Zou and Hastie 2005) linear regression (LinR) model was trained. The LinR model uses the same features listed in Tables 2 and 3 and is tuned with the values in Table 4.

Two LinR models were developed, each predicts a storm's average motion in either the x or y direction between 0 min and some lead time, at 2-min lead-time intervals. If the storm has existed for multiple time steps, current and previous (up to 10 min in the past) motion vector forecasts for a lead time are averaged to smooth changes in the storm's motion. The forecasted motion vectors are then applied at each lead-time interval to attain the storm's predicted track. Unlike binary classification models that output a probability, the LinR model does not provide any form of “uncertainty” in its prediction. However, as lead time increases, the likelihood of the LinR model incorrectly forecasting the storm's motion grows. Therefore, to issue warnings, a method must be identified to determine the uncertainty in the track and ultimately the size of a warning polygon. Again, a cost-benefit problem exists as increasing the spatial size of warnings improves the probability of a tornado landing in the warning polygon, but may result in unnecessarily large warning areas.

A width around the forecasted track was chosen such that at each lead-time interval a prescribed percentage of tornadoes in the training set were inside the warning polygon. At 2 min of lead time, the width of a warning is required to encompass 95% of the tornadoes in the training set. Across the eight iterations, this results in an average warning radius of 1.6 km from a storm's projected track. The required percentage is reduced at longer lead times such that the warning radius increases, but does not become unnecessarily large. The warning threshold decreases linearly to a value of either 50%,

70%, or 90% at 60 min of lead time resulting in a mean maximum warning width of 8.3 km (Fig. 12a), 11.2 km (Fig. 12b), or 19.8 km (Fig. 12c), respectively. The effects of these different warning widths on warning performance are examined in section 5.

Compared to section 3, we note that the development of the spatial location of the warnings has received less attention here. One reason is that there exist far more trackable storms (10^4) than tornadoes (10^2), thus allowing a simple model (e.g., LinR) to fit the dataset well enough to attain reasonable storm tracks. Nonetheless, there exist many avenues for improving the storm tracks and spatial location of the warnings. Examples include testing other types of ML models (e.g., decision trees, neural networks, etc.) and better treatment of uncertainty in the storm track.

5. Combined output

Warnings are issued within the 2D field of each simulation by combining the temporal and spatial components of the warning system. Warning polygons are issued at each time step of the simulation. The duration of the warning at each grid point is based on the temporal component of the warning system. If the warning time at a grid point reaches 0, the warning for that grid point expires. If at any time a warning of a higher temporal duration is issued for a grid point, the warning time for the grid point is updated to that value (e.g., if a warning is set to expire in 10 min, but an updated warning is issued for 20 min the warning time for the grid point will be updated to 20 min). All warnings, regardless of the warning interval, use 2-min LinR output for the spatial component of the warning.

a. System performance

POD, POD10, and LTA are used to evaluate the combined output of the warning systems (Fig. 13). Additionally, because false alarms need to be considered beyond the temporal component (which is considered through FATR), a new metric that considers the spatial extent of warnings is introduced. This metric follows the form of the false alarm ratio and FATR, such that we define a false alarm area ratio (FAAR) as

$$\text{FAAR} = \frac{\text{GFW}}{\text{GTW}}, \quad (14)$$

where GFW is the number of grid points falsely warned and GTW is the total number of grid points warned. Thus a perfect FAAR score is 0. A tornado is a small spatial event, thus we allow a buffer around each tornadic grid point such that warned grid points within the

FIG. 11. Schematic representation of genetic algorithm optimization. Yellow boxes are threshold values. Green values are SOF scores for the threshold set. A threshold set is a single row of threshold values and a SOF score.

buffer are counted as hits instead of false alarms. For a 5-km buffer,

$$FAAR_5 = \frac{GFW_5}{GTW}, \quad (15)$$

where GFW_5 is the number of warned grid points not within 5 km of a tornado. Similarly, $FAAR_{15}$ uses the same definition but allows for warned grid points within 15 km of a tornadic grid point to be counted as hits. Note, to be considered a hit, a grid point must be warned prior to the nearby grid point (i.e., the one for which a buffer is being drawn around) becoming tornadic.

Additionally, grid points where warnings have expired as false alarms are allowed to be converted to hits for up to 15 min after expiration.

Additionally, we note that our definition of a tornado has been modified in this section, such that tornadic grid points spatially separated by less than 2 km and occurring within one output time step are clustered into single tornado events. This modification is to prevent tornadic grid points separated by only a couple of grid points from being counted as separate hits/misses when in real life these would likely be classified as a single tornado. This definition results in 756 tornadoes across the 128 simulations.

FIG. 12. Example warnings paths generated by the spatial component of the warning system. Each column corresponds to a different warning width. Where the warning width is determined by the percentage of tornadoes in the training set that must occur within the warning polygon at different lead-time intervals. At 60 min of lead time, the warning polygon must contain (a) 50%, (b) 70%, or (c) 90% of the tornadoes. The maximum warning time for each grid point is colored. Black lines indicate the expected track of the storm object. The storm object location is represented by the light blue circle.

In general, the system performs well across all warning metrics (Fig. 13). As a baseline for comparison, NWS performance is noted on each graph (orange and black lines). The NWS values are based on a 3-yr average between 2016 and 2018 and are collected from the NWS Performance Management website (<https://verification.nws.noaa.gov/services/public/index.aspx>). NWS values of POD, POD10, and LTA are calculated as defined above. The AI systems' FATRs are calculated from a storm based, temporal-only perspective (i.e., FATR only evaluates the green side of Fig. 1) and thus direct comparison to a similar NWS metric is not easily performed. However, an estimate of NWS FATR calculated from a spatial and temporal perspective is $\sim 93\%$. Similarly, the calculation of FAAR for the NWS requires information about the size and exact track of a tornado. Thus, FATR and FAAR should only be used for intercomparison of the AI systems. The reason for presenting the NWS metrics is not to compare our performance, rather it provides a baseline for evaluating the skill of the AI system. Undoubtedly, performance of the automated warning system would be degraded if it was required to operate with a real-life dataset. Therefore, if the warning system's skill in this highly idealized framework were, in fact, lower than real-life NWS performance, it is unlikely that a real-life application of the AI system would be feasible.

POD is similar across the warning systems with confidence interval values generally between 80% and 90%. POD10 tends to increase as the warning interval length is increased. The increase likely corresponds to a tornado having a higher chance of occurring within a 30-min warning block than a 2-min warning block, particularly at lead times beyond 10 min. While the large warning interval could aid the POD term, the false alarm terms (e.g., FATR, FAAR) are negatively impacted by the longer warning intervals. The 30-min warning systems' confidence intervals of $1 - \text{FATR}$ are centered near 0.1 meaning that a tornado is occurring 10% of the time the system expects a tornado to occur. The 2-min systems increase this value by approximately 50%. FAAR is also negatively impacted by the longer warning interval in the RF and GBDT systems with the number of grid points classified as hits in the 2-min systems approximately 5–10 percentage points higher than in the 30-min systems. Differences in LTA are less clear with confidence intervals generally centering near 20 min. In general, differences between the RF and GBDT systems are not obvious. LR performance is at or slightly above RF and GBDT performance for POD and POD10, but LR generally performs worse with regards to the false alarm statistics.

The above results indicate that there are several solutions in issuing warnings. For example, one could

FIG. 13. Mean and 95% confidence intervals of warning metrics related to the warning system. The orange (black for POD) line represents NWS average performance between 2016 and 2018. No NWS value for FATR is shown as system values of FATR are calculated using only the temporal component of the warning system. Columns correspond to a specific system, with the ML model type and warning interval identified along the bottom. XG corresponds to gradient boosted decision trees.

have a higher POD at the cost of having more false alarms (lower FATR and FAAR). This trade-off is also noted in real-life warning performance as changing warning thresholds results in changes to warning output, but does not necessarily change the skill of the forecast (Brooks 2004; Brooks and Correia 2018). The ability to choose from different warning systems that are skilled in

different performance metrics is a benefit of an automated warning system that is discussed in more detail in section 5c. For now, we will focus on using only the 2-min 30-model RF system as it performs well with false alarm and lead-time metrics while also having a moderately high POD.

b. Three examples of system output

Warning system output was generated for three simulations (Figs. 14–16). The temporal component uses the 2-min, 30-model RF output and the spatial component uses the 2-min, 30-model LinR output. Note, the cities in Figs. 14 and 15 are identified only to provide a reference to the spatial scale of the convection and the warnings.

The first two examples show only a subset of the domain and focus on two different storm modes, a pair of semidiscrete supercells (Fig. 14) and a disorganized linear system (Fig. 15). The supercell case is notably more tornadic with each storm producing multiple tornadoes lasting several minutes. The linear storm system also produces multiple tornadoes, but they are much shorter in length and duration, each lasting less than 4 min.

In both cases, discontinuous warnings are issued (Fig. 14 at 0004 UTC and Fig. 15 at 0000 UTC) and result in increased warning lead time for some of the warned grid points. Additionally, in the cellular case one warning is issued more than 30 min in advance of the storm object center with a large amount of unwarned area between the storm and the warning (Fig. 14 at 0012 UTC near Norman). Differences also exist between the two storm modes. Most notably, the temporal lengths of the warnings issued in the cellular case (Fig. 14) are generally longer, with some of the warnings extending beyond 30 min in duration. Shorter (in temporal length) warnings are issued in the linear case (Fig. 15) with the entire warning process occurring in less than 30 min. The linear case also shows an example of a missed event. The AI system correctly issues a warning in advance of a tornado at 0016 UTC, but fails to identify a separate tornadic circulation at 0012 UTC until after the tornado has already started. The missed event shows room for improvement, but the ability of the AI system to provide 16 min of lead time for the tornado event at 0016 UTC for a “nonclassical” tornadic storm is encouraging.

Figure 16 presents a final example of the warning system where the entire model domain is visible. This example is particularly interesting because of the transition that occurs in the storm mode from linear to cellular. At 2300 UTC, a linear convective system is present and the warning system issues a warning in advance of a bowing segment (the storm in Fig. 9a). Between 2316 and 0000 UTC the warning system issues no additional

FIG. 14. Localized warning output from the AI system for a cellular case. Colored contours indicate the amount of time until a warning expires for a particular grid point. Colored triangles are similar to the black triangles used in Fig. 7 to identify grid points where a tornado has occurred, except now they have been colored to identify the amount of lead time provided by the warning for that grid point. Purple arrows indicate the discontinuous warnings referenced in the text. Blue arrows identify the area referenced in the text where a large unwarned area exists between the storm center and the warning. The blue dot represents the storm center. Note, times are not at equal intervals and instead highlight “interesting events” (e.g., tornadoes occurring, new warning polygons, etc.). Cities are identified to provide a reference to the spatial scale of the warnings. The entire animation is available as Fig. S6.

warnings and no tornadoes occur. At 0000 UTC, the warning system issues a warning less than 2 min in advance of a brief tornado. Note, that at this time a transition in storm mode is occurring with several cellular storms developing ahead of the linear convective system. By 0034 UTC one of the preline storms has developed into a supercell (the storm in Fig. 9b) and the warning system correspondingly issues a warning in advance of the storm.

As is seen in real life, warning predictability seemingly decreases in nonclassical reflectivity patterns. The AI system is able to handle these different storm modes by adjusting the length of warnings being issued. Given the added uncertainty of tornadogenesis in linear systems at long lead times, the warning system reduces the warning length and makes warning decisions closer to the time of the tornado event. Conversely, when dealing with a supercell it is more confident and

FIG. 15. As in Fig. 14, but for a linear storm system. Colored contours have been replaced with only the outline of the warning so that the background reflectivity fields are visible. The entire animation is available as Fig. S7.

increases the length of its warnings resulting in an increase in lead time. The ability to modify the length of warnings by analyzing variations in the temporal probabilities of a storm's tornado potential is an ability of an AI system that could reduce false alarm rates, both spatially and temporally. Further discussion

on overall system performance in different storm modes is discussed in [appendix C](#).

c. Changing desired performance

We have shown that variations in system performance occur by changing the warning interval and ML model

FIG. 16. As in Fig. 14, but now the entire domain is visible. The entire animation is available as Fig. S8.

type. However, other factors of the warning system can be modified to change the performance.

A simple example is to change the width of the warnings. A small warning size (Figs. 12a and 17, column 2) results in slight reductions to POD, POD10, LTA, and FAAR₅. Conversely, a larger warning size (Figs. 12c and 17, column 3) results in slight increases to these performance metrics. Note, FATR is unaffected as its value only considers the temporal component of the warnings. Additionally, FAAR₁₅ is mostly unaffected by the warning size as the 15-km buffer is wider than the width of the largest warning even at a lead time of 60 min.

The above results are expected as increasing the size of a warning increases the probability of detecting a tornado, but also increases the amount of area falsely warned (i.e., FAAR₅). Additionally, the definition of the SOF is an important component of the warning system. Changing the SOF results in changes to the optimized warning thresholds. The SOF allows the user to specify those metrics that are considered most important. Ultimately, what is considered most important is beyond the scope of this paper, but we present two

different types of SOF functions based on recent literature related to public response to tornado warnings.

Trainor et al. (2015) found that false alarm rates could affect the public's response to tornado warnings. So to reduce false alarms in the warning system, (10) is modified so that

$$\text{MCSI} = [0.25(\text{POD}_{\text{LT}})^{-1} + 1.75(1 - \text{FATR})^{-1} - 1]^{-1}. \quad (16)$$

The change to MCSI weights the false alarm term (FATR) more heavily. Additionally, (11) is modified so that

$$\text{MPMF} = \frac{\text{POD}_{\text{LT10}} - 0.30}{0.10}. \quad (17)$$

The change to MPMF requires fewer warnings to be issued at advanced lead times during warning threshold optimization. The changes to MCSI and MPMF result in drastic changes to system performance (Fig. 17, column 4). POD, POD10, and LTA are all reduced. Most notably

FIG. 17. As in Fig. 13, but now using different system optimization functions (SOFs). The “control” column is the same as column 1 in Fig. 13. WS is warning size.

though, FATR and FAAR are greatly reduced corresponding to a reduction in the amount of false alarms issued by the warning system, both temporally and spatially.

Lim et al. (2019) found that the false alarm problem, particularly in the southeastern United States, “may be overblown,” and that people continue to respond to warnings even if they have been exposed to false alarms. Based on these findings, it would seem the system should put less emphasis on minimizing false alarm terms and instead focus on increasing detection and lead time. Again, we modify two terms beginning with (9) so that

$$\text{POD}_{\text{LT}} = \frac{\sum_{i=1}^{T_{\text{wa}}} \left(\frac{\text{LT}_i}{60} \right)}{T}. \quad (18)$$

Changing the denominator in the summed term from 30 to 60 results in warnings with lead times greater than 30 min contributing to increased skill in the system. Additionally, (12) is modified, such that

$$\text{POD}_{\text{LT10}} = \frac{\sum_{i=1}^{T_{\text{wa}}} \left(\frac{\text{LT}_i}{20} \right)}{T}. \quad (19)$$

As before, each of the summed values is capped at 1. Changing the denominator value in the summed term from 10 to 20 results in an increased emphasis on warning lead times near 20 min. As expected, LTA increases to values of over 25 min (Fig. 17, column 5). To obtain this lead time, warning thresholds need to be reduced, thus resulting in an increase to POD10 while also increasing FATR, FAAR₅, and FAAR₁₅. Thus, the system has attained the goal of increasing lead time at the cost of increasing false alarms.

6. Conclusions

This study has developed a prototype framework for issuing tornado warnings using an AI system. The automated warning system developed in this paper offers several benefits when issuing warning products. First, an AI system is capable of issuing warnings at high temporal and spatial resolutions by assessing the uncertainty and variability of a storm’s tornadic potential. Second, the AI system is able to ingest and interpret large amounts of data and offer a consistent answer in the warning decision process. This benefit will become increasingly important as rapidly updating storm-scale models (e.g., Warn on Forecast; Stensrud et al. 2009, 2013; Lawson et al. 2018) become commonplace and continue to increase the information content available

to forecasters. Finally, the answers output by an AI system can be tuned to optimize a user's preferred performance metrics. Ideally, this aspect of the system could allow for different warning philosophies to be used with the "flip of a switch." For example, for a day on which violent tornadoes were expected, the system could be configured to increase warning lead times. Likewise, for days on which the tornado risk is low, the system could be configured to issue shorter warnings that avoid unnecessarily large warning areas.

Although the warning system generally performs well, there is still room for improvement. Notably, a trade-off occurs in warning metrics (e.g., as POD increases, false alarms increase) with most of the systems presented in this paper having a POD of over 80%, but still issuing false alarms nearly 85% of their warning time. Additional items have been noted throughout the paper that could result in better performance of the system. Some examples include testing different ML model types, adjusting the handling of storm track uncertainty, and adjusting optimization methods.

This study only presents a framework for an automated system to issue tornado warnings. This study has shown that skillful, automated tornado warning guidance is possible when the "observations" are essentially perfect. Real-time automated tornado warning guidance will require overcoming several hurdles including issues related to radar data (e.g., data horizons, cones of silence, regions of low signal-to-noise ratio, variable resolution, the measurement of only a single velocity component, etc.), model data (e.g., coarse model output, model forecast errors, etc.), and inaccuracies in the tornado reports database such as unreported tornadoes or strong straight-line winds classified as tornadoes (e.g., Ray et al. 2003; Doswell 2007; Potvin et al. 2019). Given these challenges, early implementation of an AI warning system may only have sufficient skill to provide real-time guidance, nonetheless, future improvement to observational and modeling capabilities may allow full automation to be explored in the future.

Acknowledgments. We appreciate conversations with Dr. Corey Potvin, Dr. Erik Rasmussen, Dr. David Stensrud, Branden Katona, and Shawn Murdzek that have benefited this work. Also, constructive comments from three anonymous reviewers helped to improve this manuscript. Additionally, we thank Dr. George Bryan for his ongoing support of CM1 and Dr. Harold Brooks for his assistance in accessing NWS warning performance statistics. Data for this project are available through the Penn State data commons. This work was supported by NOAA Award NA18OAR4590310.

APPENDIX A

Storm Mode Distributions

A *K*-means clustering was performed using 20 km $\times$ 20 km images of reflectivity at a height of 500 m centered on each storm object (Fig. A1). An "elbow test" of the within-cluster sum-of-squares error, resulted in the choice to use five clusters. As with any clustering algorithm, some overlap exists between clusters, but distinct storm mode characteristics are apparent. For example, cluster 5 exhibits discrete supercell characteristics. Conversely, cluster one is more linear and disorganized. Clusters 2, 3, and 4 show mixed-mode characteristics with cluster 2 favoring a more linear structure, cluster 3 exhibiting a nondiscrete supercell-like structure, and cluster four falling somewhere in between. Clusters 1 and 2 contain the highest percentage of storm objects. This result is expected and intended as storms were initiated using methods to encourage upscale, linear growth. The largest percentage of tornadoes form in storms assigned to cluster three (33.9%).

APPENDIX B

Genetic Algorithm Optimization

To begin GA optimization, a parent population of 500 threshold sets are created. Each threshold set contains a threshold value for each warning lead time in the system (e.g., the 2-min system has 30 threshold values in each of the 500 threshold sets). The threshold values in each of the 500 sets are randomly assigned a value between 0.4 and 1.06. If the probability from a ML model is greater than the threshold at a particular lead time, a warning is issued. Note, warning thresholds are allowed to exceed 1, and a threshold greater than 1 makes it impossible for a warning to be issued for that lead time.

GA makes use of three techniques to adjust the threshold values: selection, crossover and mutation. The optimization procedure begins with selection where three threshold sets are randomly chosen from the population and their SOF values are compared. Among the three options, the threshold set with the largest SOF value is advanced to the offspring population. This selection process is repeated 500 times. Because the size of the parent population (500) equals the offspring population (500), threshold sets from the parent population can be selected more than once. After creating the initial offspring population, crossover and mutation techniques, similar to those developed in Deb et al. (2002), are applied. In the crossover operation, two sets in the offspring population are randomly chosen and their

FIG. A1. A K -means clustering of 500-m simulated reflectivity. Each set of three images is a cluster with the cluster label to the left of the images. (from left to right) The images represent the cluster center, the storm object experiencing a tornadogenesis event with the smallest Euclidean distance to the cluster center, and the storm object experiencing a tornadogenesis event with median Euclidean distance to the cluster center among storm objects labeled to that cluster. The numbers in the blue (red) boxes indicate the percentage of storm objects (tornadogenesis events) assigned to each cluster.

threshold values are blended. The threshold values chosen for blending (P_1 and P_2) produce two children (C_1 and C_2) where

$$C_1 = 0.5[P_1 + P_2 - \beta(P_2 - P_1)], \quad (\text{B1})$$

and

$$C_2 = 0.5[P_1 + P_2 + \beta(P_2 - P_1)], \quad (\text{B2})$$

where β depends on α , which is defined as

$$\alpha = 2.0 - \kappa^{-(\eta+1)}, \quad (\text{B3})$$

where η is 0.75 and

$$\kappa = 1.0 + 2.0 \left(\frac{P_1 - 0.4}{P_2 - P_1} \right), \quad (\text{B4})$$

for C_1 and

$$\kappa = 1.0 + 2.0 \left(\frac{1.06 - P_2}{P_2 - P_1} \right), \quad (\text{B5})$$

for C_2 . Finally, a random number between 0 and 1 is defined as r , so that

$$\beta = \begin{cases} \frac{1}{(r\alpha)^{\eta+1}}, & r \leq \frac{1}{\alpha} \\ \left(\frac{1}{2 - r\alpha} \right)^{\frac{1}{\eta+1}}, & r > \frac{1}{\alpha} \end{cases}. \quad (\text{B6})$$

As an example, if parent threshold values of 0.80 and 0.90 were chosen to blend, examples of their children could be 0.82 and 0.88, 0.75 and 0.95, etc. Mutation is performed by randomly changing 20% of the threshold values in a threshold set. The chance that a threshold set within the population is chosen to be crossed or mutated is 50% and 20%, respectively.

APPENDIX C

Performance by Storm Mode

The 2-min, 30-model RF system was evaluated by cluster (Fig. C1). For POD, POD10, and LTA, each tornado is assigned a cluster label based on the cluster label of the nearest storm object at the time step of the tornado's formation. For FATR, the cluster of the false

alarm/hit is assigned when a warning expires or a tornado is produced for a particular storm at a particular time step. FAAR is not calculated as it requires assigning cluster labels to grid points, which is nontrivial as storms can change clusters and thus assign different cluster values to the same grid point. Certainly, methods could be developed to handle this calculation, but we leave this item as a point of future work. Finally, extreme caution should be taken in the interpretation of these results. By breaking the performance down by cluster, we are reducing the number of data points that are used to calculate the statistics in each of the eight folds. This situation is particularly evident for cluster 5 (which contains only 3.6% of the tornado events) where the confidence intervals are so large they provide little room for interpretation, and thus are not discussed below.

In general, clusters 1–4 perform as expected (Fig. C1). Cluster 1, which resembles disorganized convection, performs the worst with regard to POD and LTA. The warning system predicts ~70% of the tornadoes in cluster 1, but lead time is reduced by ~10 min compared to the full population of storms. Surprisingly, FATR in cluster 1 is on par with the general performance of the system. As discussed in section 5b, in nonclassical convection the warning system tends to shorten its warnings, thus reducing the time and area of false alarms. This “situational awareness” of the AI system, may be the reason it avoids a high FATR in the disorganized convection of cluster 1. Cluster 2, which still contains linear structures, but appears slightly more organized, performs better than cluster 1. In fact, the performance of Cluster 2 almost matches the general performance of the system. Cluster 3 has the most cellular structure (aside from cluster 5) and the best performance. Most notably, false alarms are reduced and lead time is increased. The likely reason for cluster 3’s top performance is twofold: 1) cellular storms are easier to forecast and 2) cluster 3 contains the highest percentage of tornadoes and thus the ML models are learning on a higher percentage of storms with similar characteristics resulting in above average performance. Finally, cluster 4 experiences poorer performance than clusters 2 and 3, particularly with regards to FATR. The reason for this difference is unclear; however, it is again worth noting the percentage of tornadoes assigned to cluster 4 is much less than clusters 2 and 3, and thus the reason may simply be the ML models having less training on the storm type in cluster 4.

Comparison of the above results to NWS performance is made difficult by the need to classify the convective mode of real-life storms. This task is beyond the scope of this project; however, Brotzge et al. (2013) did study the

FIG. C1. As in Fig. 13, but now for each of the clusters in Fig. A1.

performance of tornado warnings by storm mode between 2003 and 2004. The NWS performance by storm mode found in Brotzge et al. (2013) is summarized in their Table 2. Both the AI system and the NWS experience lower POD and lead time in linear and disorganized storms. Compared to NWS performance, the AI

system increases POD in linear/disorganized storms and increases lead time in cellular storms. Additional caution should be taken as these comparisons are done without consideration of false alarms. Furthermore, the average NWS POD has dropped (in association with a reduction in false alarms) between 2003–04 (~70%) and 2016–18 (~50%) by ~20 percentage points (Brooks and Correia 2018). Thus, the values in Brotzge et al. (2013) may overestimate NWS performance by storm mode in comparison to the 2016–2018 NWS performance presented in section 5a.

REFERENCES

- Adrianto, I., T. Trafalis, and V. Lakshmanan, 2009: Support vector machines for spatiotemporal tornado prediction. *Int. J. Gen. Syst.*, **38**, 759–776, <https://doi.org/10.1080/03081070601068629>.
- Brooks, H. E., 2004: Tornado-warning performance in the past and future: A perspective from signal detection theory. *Bull. Amer. Meteor. Soc.*, **85**, 837–844, <https://doi.org/10.1175/BAMS-85-6-837>.
- , and J. Correia, 2018: Long-term performance metrics for national weather service tornado warnings. *Wea. Forecasting*, **33**, 1501–1511, <https://doi.org/10.1175/WAF-D-18-0120.1>.
- Brotzge, J. A., S. E. Nelson, R. L. Thompson, and B. T. Smith, 2013: Tornado probability of detection and lead time as a function of convective mode and environmental parameters. *Wea. Forecasting*, **28**, 1261–1276, <https://doi.org/10.1175/WAF-D-12-00119.1>.
- Bryan, G. H., and J. M. Fritsch, 2002: A benchmark simulation for moist nonhydrostatic numerical models. *Mon. Wea. Rev.*, **130**, 2917–2928, [https://doi.org/10.1175/1520-0493\(2002\)130<2917:ABSFMN>2.0.CO;2](https://doi.org/10.1175/1520-0493(2002)130<2917:ABSFMN>2.0.CO;2).
- Chen, T., and C. Guestrin, 2016: XGBoost: A scalable tree boosting system. *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, ACM, 785–794, <https://doi.org/10.1145/2939672.2939785>.
- Cintineo, J. L., M. J. Pavolonis, J. M. Sieglaff, and D. T. Lindsey, 2014: An empirical model for assessing the severe weather potential of developing convection. *Wea. Forecasting*, **29**, 639–653, <https://doi.org/10.1175/WAF-D-13-00113.1>.
- , and Coauthors, 2018: The NOAA/CIMSS probsevere model: Incorporation of total lightning and validation. *Wea. Forecasting*, **33**, 331–345, <https://doi.org/10.1175/WAF-D-17-0099.1>.
- Clark, A. J., and Coauthors, 2012: An overview of the 2010 Hazardous Weather Testbed experimental forecast program spring experiment. *Bull. Amer. Meteor. Soc.*, **93**, 55–74, <https://doi.org/10.1175/BAMS-D-11-00040.1>.
- Coleman, T. A., and K. J. Pence, 2009: The proposed 1883 Holden tornado warning system. *Bull. Amer. Meteor. Soc.*, **90**, 1789–1796, <https://doi.org/10.1175/2009BAMS2886.1>.
- , K. R. Knupp, J. Spann, J. B. Elliott, and B. E. Peters, 2011: The history (and future) of tornado warning dissemination in the United States. *Bull. Amer. Meteor. Soc.*, **92**, 567–582, <https://doi.org/10.1175/2010BAMS3062.1>.
- Deardorff, J. W., 1980: Stratocumulus-capped mixed layers derived from a three-dimensional model. *Bound.-Layer Meteor.*, **18**, 495–527, <https://doi.org/10.1007/BF00119502>.
- Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan, 2002: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, **6**, 182–197, <https://doi.org/10.1109/4235.996017>.
- Doswell, C. A., III, 2007: Small sample size and data quality issues illustrated using tornado occurrence data. *Electron. J. Severe Storms Meteor.*, **2** (5), <https://www.ejssm.org/ojs/index.php/ejssm/article/viewArticle/26/27>.
- , S. J. Weiss, and R. H. Johns, 1993: Tornado forecasting: A review. *The Tornado: Its Structure, Dynamics Prediction, and Hazards*, Geophys. Monogr., Vol. 79, <https://doi.org/10.1029/GM079p0557>.
- Flora, M. L., C. Potvin, P. Skinner, and A. McGovern, 2020: Using machine learning to improve storm-scale 1-h probabilistic forecasts of severe weather. *19th Conf. on Artificial Intelligence for Environmental Science*, Boston, MA, Amer. Meteor. Soc., 3B.4, <https://ams.confex.com/ams/2020Annual/webprogram/Paper367791.html>.
- Fortin, F.-A., F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, 2012: DEAP: Evolutionary algorithms made easy. *J. Mach. Learn. Res.*, **13**, 2171–2175.
- Gagne, D. J., A. McGovern, S. E. Haupt, R. A. Sobash, J. K. Williams, and M. Xue, 2017: Storm-based probabilistic hail forecasting with machine learning applied to convection-allowing ensembles. *Wea. Forecasting*, **32**, 1819–1840, <https://doi.org/10.1175/WAF-D-17-0010.1>.
- Galway, J. G., 1989: The evolution of severe thunderstorm criteria within the weather service. *Wea. Forecasting*, **4**, 585–592, [https://doi.org/10.1175/1520-0434\(1989\)004<0585:TEOSTC>2.0.CO;2](https://doi.org/10.1175/1520-0434(1989)004<0585:TEOSTC>2.0.CO;2).
- Goldberg, D. E., 1989: *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st ed. Addison-Wesley Longman Publishing Co., Inc., 372 pp.
- Holden, E. S., 1883: A system of local warnings against tornadoes. *Science*, **ns-2**, 521–522, <https://doi.org/10.1126/SCIENCE.NS-2.37.521>.
- Holland, J. H., 1975: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, 211 pp.
- James, P. M., B. K. Reichert, and D. Heizenreder, 2018: NowCastMIX: Automatic integrated warnings for severe convection on nowcasting time scales at the German Weather Service. *Wea. Forecasting*, **33**, 1413–1433, <https://doi.org/10.1175/WAF-D-18-0038.1>.
- Kain, J. S., P. R. Janish, S. J. Weiss, M. E. Baldwin, R. S. Schneider, and H. E. Brooks, 2003: Collaboration between forecasters and research scientists at the NSSL and SPC: The spring program. *Bull. Amer. Meteor. Soc.*, **84**, 1797–1806, <https://doi.org/10.1175/BAMS-84-12-1797>.
- , and Coauthors, 2008: Some practical considerations regarding horizontal resolution in the first generation of operational convection-allowing NWP. *Wea. Forecasting*, **23**, 931–952, <https://doi.org/10.1175/WAF2007106.1>.
- Karstens, C. D., and Coauthors, 2018: Development of a human-machine mix for forecasting severe convective events. *Wea. Forecasting*, **33**, 715–737, <https://doi.org/10.1175/WAF-D-17-0188.1>.
- Lagerquist, R., A. McGovern, and T. Smith, 2017: Machine learning for real-time prediction of damaging straight-line convective wind. *Wea. Forecasting*, **32**, 2175–2193, <https://doi.org/10.1175/WAF-D-17-0038.1>.
- , —, C. Homeyer, D. J. Gagne, and T. M. Smith, 2020: Short-term tornado prediction via deep learning on 3D multiscale data. *Severe Local Storms Symp.*, Boston, MA, Amer. Meteor. Soc., 3.3, <https://ams.confex.com/ams/2020Annual/meetingapp.cgi/Paper/366773>.
- Lawson, J. R., J. S. Kain, N. Yussouf, D. C. Dowell, D. M. Wheatley, K. H. Knopfmeier, and T. A. Jones, 2018:

- Advancing from convection-allowing NWP to warn-on-forecast: Evidence of progress. *Wea. Forecasting*, **33**, 599–607, <https://doi.org/10.1175/WAF-D-17-0145.1>.
- Lim, J. R., B. F. Liu, and M. Egnoto, 2019: Cry wolf effect? Evaluating the impact of false alarms on public responses to tornado alerts in the southeastern United States. *Wea. Climate Soc.*, **11**, 549–563, <https://doi.org/10.1175/WCAS-D-18-0080.1>.
- Mansell, E. R., C. L. Ziegler, and E. C. Bruning, 2010: Simulated electrification of a small thunderstorm with two-moment bulk microphysics. *J. Atmos. Sci.*, **67**, 171–194, <https://doi.org/10.1175/2009JAS2965.1>.
- Marzban, C., and G. J. Stumpf, 1996: A neural network for tornado prediction based on Doppler radar-derived attributes. *J. Appl. Meteor.*, **35**, 617–626, [https://doi.org/10.1175/1520-0450\(1996\)035<0617:ANNFTP>2.0.CO;2](https://doi.org/10.1175/1520-0450(1996)035<0617:ANNFTP>2.0.CO;2).
- , and —, 1998: A neural network for damaging wind prediction. *Wea. Forecasting*, **13**, 151–163, [https://doi.org/10.1175/1520-0434\(1998\)013<0151:ANNFDW>2.0.CO;2](https://doi.org/10.1175/1520-0434(1998)013<0151:ANNFDW>2.0.CO;2).
- , and A. Witt, 2001: A Bayesian neural network for severe-hail size prediction. *Wea. Forecasting*, **16**, 600–610, [https://doi.org/10.1175/1520-0434\(2001\)016<0600:ABNNFS>2.0.CO;2](https://doi.org/10.1175/1520-0434(2001)016<0600:ABNNFS>2.0.CO;2).
- McCauley, E. W., and M. L. Weisman, 2001: The sensitivity of simulated supercell structure and intensity to variations in the shapes of environmental buoyancy and shear profiles. *Mon. Wea. Rev.*, **129**, 664–687, [https://doi.org/10.1175/1520-0493\(2001\)129<0664:TSOSSS>2.0.CO;2](https://doi.org/10.1175/1520-0493(2001)129<0664:TSOSSS>2.0.CO;2).
- McGovern, A., D. J. Gagne, J. K. Williams, R. A. Brown, and J. B. Basara, 2014: Enhancing understanding and improving prediction of severe weather through spatiotemporal relational learning. *Mach. Learn.*, **95**, 27–50, <https://doi.org/10.1007/s10994-013-5343-x>.
- , K. L. Elmore, D. J. Gagne, S. E. Haupt, C. D. Karstens, R. Lagerquist, T. Smith, and J. K. Williams, 2017a: Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bull. Amer. Meteor. Soc.*, **98**, 2073–2090, <https://doi.org/10.1175/BAMS-D-16-0123.1>.
- , C. Potvin, and R. Brown, 2017b: Using large-scale machine learning to improve our understanding of the formation of tornadoes. *Large-Scale Machine Learning in the Earth Sciences*, A. N. Srivastava, R. Nemani, and K. Steinhaeuser, Eds., Chapman and Hall/CRC, 95–112, <https://doi.org/10.4324/9781315371740-6>.
- , C. D. Karstens, T. Smith, and R. Lagerquist, 2019a: Quasi-operational testing of real-time storm-longevity prediction via machine learning. *Wea. Forecasting*, **34**, 1437–1451, <https://doi.org/10.1175/WAF-D-18-0141.1>.
- , R. Lagerquist, D. John Gagne, G. E. Jergensen, K. L. Elmore, C. R. Homeyer, and T. Smith, 2019b: Making the black box more transparent: Understanding the physical implications of machine learning. *Bull. Amer. Meteor. Soc.*, **100**, 2175–2199, <https://doi.org/10.1175/BAMS-D-18-0195.1>.
- Miller, R. C., and C. A. Crisp, 1999: The first operational tornado forecast twenty million to one. *Wea. Forecasting*, **14**, 479–483, [https://doi.org/10.1175/1520-0434\(1999\)014<0479:TFOTFT>2.0.CO;2](https://doi.org/10.1175/1520-0434(1999)014<0479:TFOTFT>2.0.CO;2).
- Mitchell, E. D. W., S. V. Vasiloff, G. J. Stumpf, A. Witt, M. D. Eilts, J. T. Johnson, and K. W. Thomas, 1998: The National Severe Storms Laboratory tornado detection algorithm. *Wea. Forecasting*, **13**, 352–366, [https://doi.org/10.1175/1520-0434\(1998\)013<0352:TNSSLT>2.0.CO;2](https://doi.org/10.1175/1520-0434(1998)013<0352:TNSSLT>2.0.CO;2).
- Morrison, H., J. A. Curry, M. D. Shupe, and P. Zuidema, 2005: A new double-moment microphysics parameterization for application in cloud and climate models. Part II: Single-column modeling of Arctic clouds. *J. Atmos. Sci.*, **62**, 1678–1693, <https://doi.org/10.1175/JAS3447.1>.
- , G. Thompson, and V. Tatarskii, 2009: Impact of cloud microphysics on the development of trailing stratiform precipitation in a simulated squall line: Comparison of one- and two-moment schemes. *Mon. Wea. Rev.*, **137**, 991–1007, <https://doi.org/10.1175/2008MWR2556.1>.
- Pedregosa, F., and Coauthors, 2011: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.
- Potvin, C. K., C. Broyles, P. S. Skinner, H. E. Brooks, and E. Rasmussen, 2019: A Bayesian hierarchical modeling framework for correcting reporting bias in the U.S. tornado database. *Wea. Forecasting*, **34**, 15–30, <https://doi.org/10.1175/WAF-D-18-0137.1>.
- Ray, P. S., P. Bieringer, X. Niu, and B. Whissel, 2003: An improved estimate of tornado occurrence in the central plains of the United States. *Mon. Wea. Rev.*, **131**, 1026–1031, [https://doi.org/10.1175/1520-0493\(2003\)131<1026:AIEOTO>2.0.CO;2](https://doi.org/10.1175/1520-0493(2003)131<1026:AIEOTO>2.0.CO;2).
- Schaefer, J. T., 1990: The critical success index as an indicator of warning skill. *Wea. Forecasting*, **5**, 570–575, [https://doi.org/10.1175/1520-0434\(1990\)005<0570:TCSIAA>2.0.CO;2](https://doi.org/10.1175/1520-0434(1990)005<0570:TCSIAA>2.0.CO;2).
- Stensrud, D. J., and Coauthors, 2009: Convective-scale warn-on-forecast system. *Bull. Amer. Meteor. Soc.*, **90**, 1487–1500, <https://doi.org/10.1175/2009BAMS2795.1>.
- , and Coauthors, 2013: Progress and challenges with warn-on-forecast. *Atmos. Res.*, **123**, 2–16, <https://doi.org/10.1016/J.ATMOSRES.2012.04.004>.
- Stumpf, G. J., A. Witt, E. D. Mitchell, P. L. Spencer, J. T. Johnson, M. D. Eilts, K. W. Thomas, and D. W. Burgess, 1998: The National Severe Storms Laboratory mesocyclone detection algorithm for the WSR-88D. *Wea. Forecasting*, **13**, 304–326, [https://doi.org/10.1175/1520-0434\(1998\)013<0304:TNSSLM>2.0.CO;2](https://doi.org/10.1175/1520-0434(1998)013<0304:TNSSLM>2.0.CO;2).
- Trainor, J., D. Nagele, B. Philips, and B. Scott, 2015: Tornadoes, social science, and the false alarm effect. *Wea. Climate Soc.*, **7**, 333–352, <https://doi.org/10.1175/wcas-d-14-00052.1>.
- Weisman, M. L., and J. B. Klemp, 1982: The dependence of numerically simulated convective storms on vertical wind shear and buoyancy. *Mon. Wea. Rev.*, **110**, 504–520, [https://doi.org/10.1175/1520-0493\(1982\)110<0504:TDONSC>2.0.CO;2](https://doi.org/10.1175/1520-0493(1982)110<0504:TDONSC>2.0.CO;2).
- Witt, A., M. D. Eilts, G. J. Stumpf, J. T. Johnson, E. D. W. Mitchell, and K. W. Thomas, 1998: An enhanced hail detection algorithm for the WSR-88D. *Wea. Forecasting*, **13**, 286–303, [https://doi.org/10.1175/1520-0434\(1998\)013<0286:AEHDAF>2.0.CO;2](https://doi.org/10.1175/1520-0434(1998)013<0286:AEHDAF>2.0.CO;2).
- Ziegler, C. L., 1985: Retrieval of thermal and microphysical variables in observed convective storms. Part I: Model development and preliminary testing. *J. Atmos. Sci.*, **42**, 1487–1509, [https://doi.org/10.1175/1520-0469\(1985\)042<1487:ROTAMV>2.0.CO;2](https://doi.org/10.1175/1520-0469(1985)042<1487:ROTAMV>2.0.CO;2).
- Zou, H., and T. Hastie, 2005: Regularization and variable selection via the elastic net. *J. Roy. Stat. Soc. Series B Stat. Methodol.*, **67**, 301–320, <https://doi.org/10.1111/j.1467-9868.2005.00503.x>.